

[WEDT.A] Dokumentacja końcowa projektu

Klasyfikacja stron WWW na podstawie struktury

Michał Aniserowicz <michalaniserowicz@gmail.com>
Jakub Turek jkbturek@gmail.com

30 maja 2013r.

1 Temat projektu

Tematem projektu jest automatyczna klasyfikacja stron WWW na podstawie struktury. W ramach uściślenia tematu projektu, wybrane do rozpoznawania zostały następujące kategorie stron internetowych:

- dzienniki internetowe (*blogi*),
- strony społecznościowe oparte na obrazkach (*kwejki*),
- serwisy informacyjne,
- sklepy internetowe.

Projekt został wykonany w technologii Python 2.7.4 i był testowany na systemach operacyjnych Windows 7 oraz Ubuntu 13.04. Projekt wykorzystuje bibliotekę PIL¹ w wersji 1.1.7.

2 Implementacja

2.1 Schemat działania aplikacji

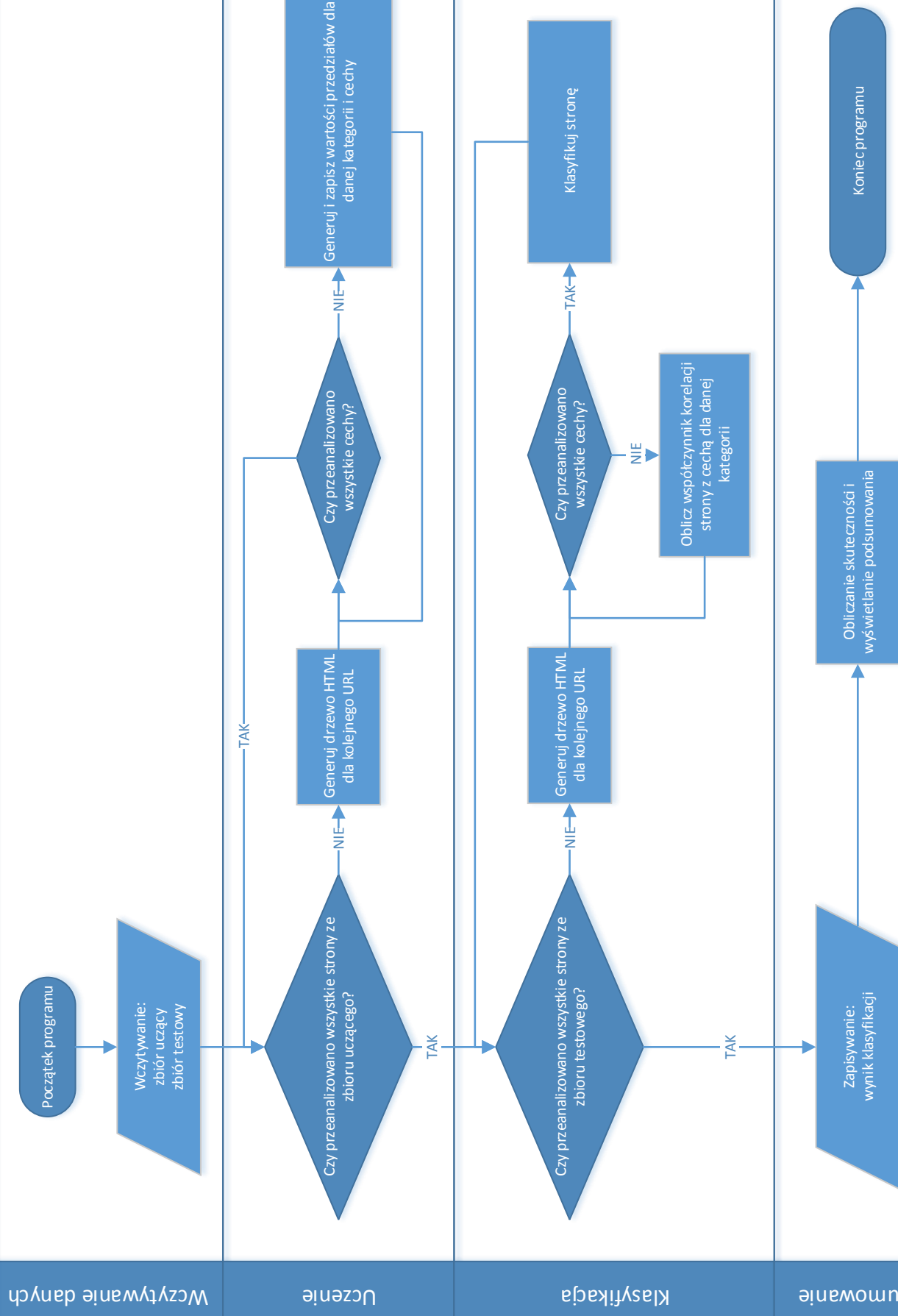
Na następnej stronie przedstawiony został ogólny schemat działania programu. Obejmuje on dwie główne fazy działania aplikacji:

uczenie się Program generuje zestawienie wartości cech dla poszczególnych kategorii na podstawie próby wzorców.

klasyfikacja Program dokonuje klasyfikacji pozostałych stron na podstawie wartości cech wyznaczonych w poprzednim kroku.

¹Python Image Library - <http://www.pythonware.com/products/pil/>.

Schemat działania programu

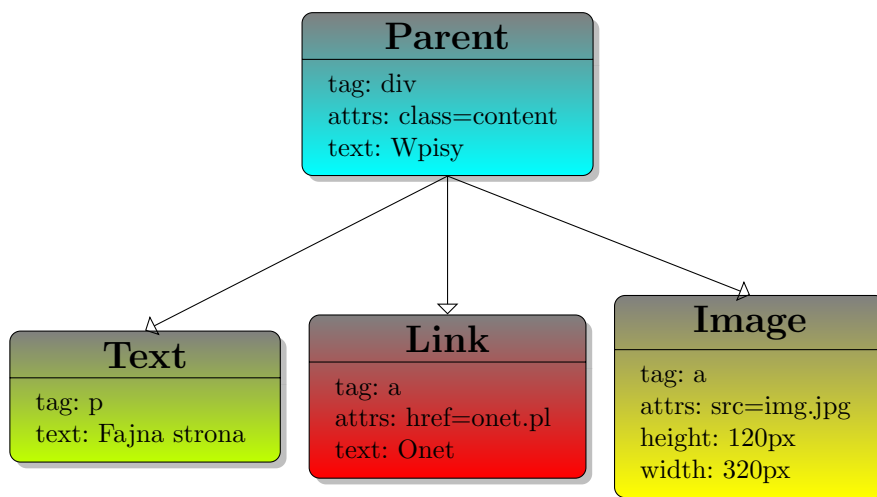


2.2 Drzewo HTML

Strony WWW są wewnętrznie reprezentowane przez drzewo HTML. Każdy korzeń drzewa posiada następujące atrybuty:

- tag,
- słownik atrybutów (np. `class="main-img"`, `src="image.jpg"`),
- tekst wewnątrz taga,
- wysokość elementu,
- szerokość elementu.

Ponadto od dowolnego korzenia można dojść do jego rodzica, a także wszystkich jego dzieci. Rysunek 1 przedstawia schemat wykorzystywanego drzewa HTML.



Rysunek 1: Poglądowy schemat drzewa HTML.

Etap budowy drzewa jest w pełni konfigurowalny, dzięki użyciu następujących atrybutów:

dozwolone tagi Lista tagów, z których mogą powstawać liście drzewa. Jeżeli tag nie znajduje się na dozwolonej liście, tekst znajdujący się w jego środku jest konkatelowany z tekstem jego pierwszego dozwolonego rodzica.

dozwolone atrybuty Lista atrybutów, które są włączane do słownika w liściach drzewa. Pozostałe atrybuty i ich wartości są pomijane.

zakazane tagi Lista tagów, które zawierają tekst nie włączany do pierwszego dozwolonego rodzica. Umożliwia to odfiltrowywanie m.in. skryptów.

Do budowy drzewa została wykorzystana wewnętrzna biblioteka języka Python - `HTMLParser`².

2.3 Główna struktura strony

Przed omówieniem zestawu cech badanych przez aplikację, należy wprowadzić pojęcie *głównej struktury strony*, która będzie intensywnie analizowana. Główna struktura strony to najliczniejsza struktura, która posiada następujące cechy:

- Występują w niej wyłącznie tagi `<td>` lub `<div>`, przy czym w danej strukturze są to tagi wyłącznie jednego z tych typów.
- Wszystkie tagi występują na jednym poziomie głębokości w drzewie. Oznacza to, że mają wspólnego rodzica.
- Każdy z tagów posiada jednakową wartość atrybutu `class`.
- Każdy element struktury posiada więcej niż jeden element podrzędny lub większy niż jeden poziom zagłębienia elementów.

Główna struktura jest zaprezentowana na rysunku 2.

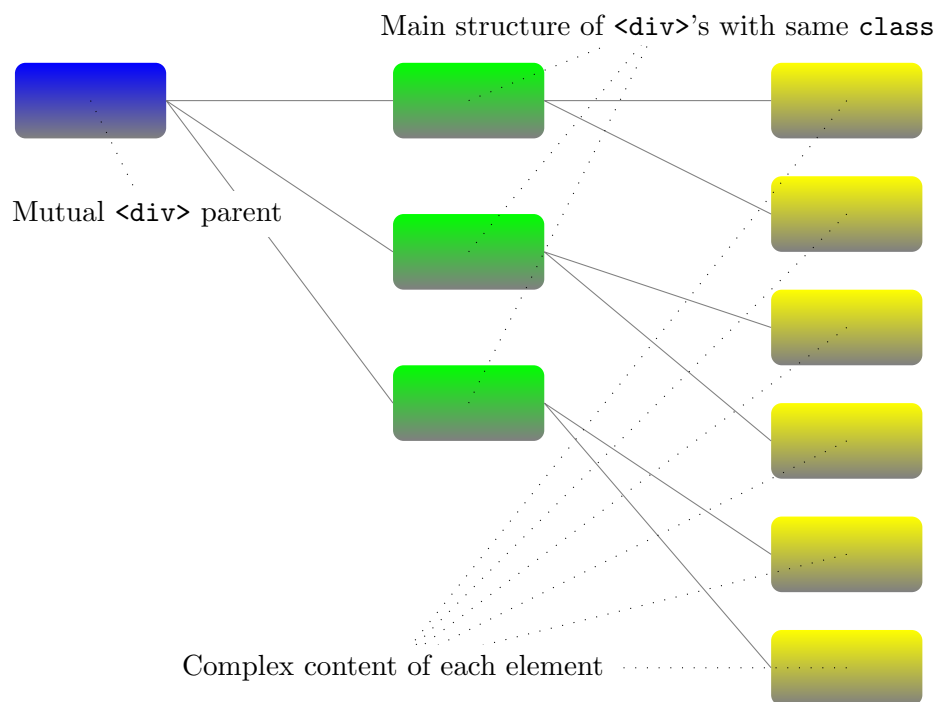
Struktura ta jest charakterystyczna dla wszystkich czterech typów klasyfikowanych stron. Na *blogach* zawiera treść wpisów, na *kwejkach* obrazki, na stronach informacyjnych odnośniki do artykułów, natomiast w sklepach internetowych - odnośniki do kategorii i/lub produktów.

2.4 Analizowane cechy strony

Na etapie uczenia się oraz klasyfikacji brane są pod uwagę, między innymi, następujące cechy:

- Liczba elementów w głównej strukturze strony.
- Liczba powtórzeń głównej struktury strony. Sprawdza czy struktura się powtarza i, jeżeli tak, to w jakiej liczbie.
- Średnia ilość tekstu przypadająca na każdy element głównej struktury strony oraz jego dzieci.
- Średnia ilość obrazów przypadająca na każdy element głównej struktury strony oraz jego dzieci.

²Dokumentacja biblioteki jest dostępna pod tym adresem.



Rysunek 2: Zielone elementy należą do głównej struktury strony.

- Liczba obrazów na stronie.
- Rozmiary największego oraz najmniejszego obrazu na stronie.
- Liczba odnośników na stronie.
- Stosunek długości tekstu zawartego w odnośnikach do długości całego tekstu zawartego na stronie.
- Średnia długość lokalnego odnośnika³ na stronie.
- Liczba tagów w specyfikacji HTML5 (przykładowo: `<article>` oraz `<section>`).
- Stosunek liczby odnośników do liczby obrazów na stronie.
- Stosunek długości tekstu do liczby obrazów na stronie.

Cechy te dobrze różnicują cztery zadane kategorie stron internetowych. Przykładowo serwisy informacyjne charakteryzują się wysokim współczynnikiem długości tekstu w odnośnikach do całkowitej długości tekstów, długimi

³Odnosnik lokalny prowadzi do tej samej domeny, w której znajduje się analizowana strona.

odnośnikami lokalnymi, dużą ilością obrazów na stronie oraz częstym występowaniem tagów w specyfikacji HTML5. Dla kontrastu blogi charakteryzują się niewielkim stosunkiem liczby odnośników do długości tekstu, niewielką liczbą obrazków, dużą koncentracją tekstu w głównej strukturze oraz większą niż dla serwisów informacyjnych liczbą elementów w głównej strukturze strony.

3 Dane

Dane wejściowe/wyjściowe są umieszczane w plikach o strukturze wewnętrznej `<adres URL>_<kategoria>`. Kolejne dokumenty są rozdzielone znakami nowej linii. Przykład struktury został zaprezentowany poniżej:

```
http://kwejk.pl kwejk
http://ataklonow.pl kwejk
http://rafalstec.blox.pl blog
http://webitect.net blog
http://onet.pl serwis informacyjny
http://wp.pl serwis informacyjny
http://wicompl.pl sklep internetowy
http://morele.net sklep internetowy
```

Rysunek 3: Struktura danych wejściowych/wyjściowych.

3.1 Dane wejściowe

Dane wejściowe składają się z dwóch plików⁴:

input_classified.txt Dane trenujące dla algorytmu. Na ich podstawie budowana jest lista kategorii stron rozpoznawanych przez system.

input_unclassified.txt Dane testowe dla algorytmu. Strony umieszczone na liście poddawane są klasyfikacji na podstawie wartości parametrów wyznaczonych przez dane trenujące. Wstępna klasyfikacja stron jest niezbędna do obliczenia skuteczności algorytmu i nie jest brana pod uwagę przez właściwy algorytm.

3.2 Dane wyjściowe

Na dane wyjściowe składają się zarówno pliki, jak również wyjście standardowe (konsola):

⁴Nazwy plików są konfigurowalne, podobnie jak inne ustawienia aplikacji, w pliku `config.ini`.

plik output.txt Wynik działania algorytmu. Zawiera adresy analizowanych stron i przyporządkowaną im przez algorytm klasyfikację.

wynik konfiguracji Wyprowadzany na wyjście standardowe. Przedstawia wartości dozwolonych przedziałów cech, w ramach kategorii, wyznaczone na podstawie danych testowych.

podsumowanie Wyprowadzane na wyjście standardowe. Przedstawia skuteczności algorytmu dla każdej kategorii, mierzone w czterech własnościach:

- dokładność,
- zupełność,
- zaszumienie,
- precyzja.

4 Skuteczność algorytmu

Skuteczność algorytmu mierzona jest przy pomocy czterech wskaźników. Zakładając, że:

- $|TP|$ to liczba poprawnych przydziałów dokumentu do danej kategorii,
- $|FP|$ to liczba niepoprawnych przydziałów dokumentu do danej kategorii,
- $|TN|$ to liczba poprawnych nieprzydziałów dokumentu do danej kategorii,
- $|FN|$ to liczba niepoprawnych nieprzydziałów dokumentu do danej kategorii,

wtedy wskaźniki te wyrażają się następującymi wzorami:

dokładność $\frac{|TP|+|TN|}{|TP|+|TN|+|FP|+|FN|},$

zupełność $\frac{|TP|}{|TP|+|FN|},$

zaszumienie $\frac{|FP|}{|FP|+|TN|},$

precyzja $\frac{|TP|}{|TP|+|FP|}.$

Wskaźniki te można odczytywać następująco:

dokładność opisuje procent poprawnych wskazań algorytmu ogółem.

zupełność opisuje procent pokrycia wejściowego zbioru danych poprawnymi wskazaniami.

zaszumienie opisuje procent właściwości wskazań negatywnych.

precyzja opisuje procent właściwości wskazań pozytywnych.

4.1 Okrojone dane testowe

Tabela 1 przedstawia wynik działania algorytmu dla okrojonych danych testowych, przygotowanych na potrzeby prezentacji. Okrojone dane testowe obejmują:

- po trzy przykłady trenujące z każdej z czterech kategorii,
- po dziesięć przykładów stron do sklasyfikowania z każdej z czterech kategorii,
- pięć stron, które nie należą do żadnej z czterech kategorii.

		Wskaźniki			
		Dokładność	Szczegółowość	Rozrzut	Precyzja
Kategorie	Blogi	98%	100%	3%	91%
	Kwejki	95%	100%	6%	83%
	Serwisy informacyjne	93%	88%	6%	78%
	Sklepy internetowe	93%	70%	0%	100%

Tabela 1: Skuteczność algorytmu dla okrojonego zestawu danych testowych.

Widać, że zaimplementowany algorytm zapewnia bardzo wysoką skuteczność. Wyróżnia się zwłaszcza skuteczność klasyfikacji blogów, dla której najgorszy z wymienionych wskaźników ma rozrzut $\pm 9\%$ względem doskonałości. Licząc średnie odchylenie wskaźników od doskonałości można wywnioskować, że algorytm jest miarodajny w 92,125% przypadków.

4.2 Pełne dane testowe

Algorytm był testowany dla następującej ilości danych:

- 108 stron z kategorii blog,
- 83 strony z kategorii kwejki,
- 67 stron z kategorii serwisy informacyjne,

- 72 strony z kategorii sklepy internetowe,
- 39 stron nie należących do żadnej z powyższych kategorii.

Wynik testów został przedstawiony w tabeli 2.

		Wskaźniki			
		Dokładność	Szczegółowość	Rozrzut	Precyzja
Kategorie	Blogi	96,75%	92,59%	1,53%	96,15%
	Kwejki	95,12%	86,75%	2,79%	90%
	Serwisy informacyjne	97,29%	92,54%	2,30%	89,86%
	Sklepy internetowe	94,85%	83,33%	3,33%	85,71%
	Inne	96,48%	89,74%	3,31%	76,09%

Tabela 2: Skuteczność algorytmu dla pełnego zestawu danych testowych.

Ponownie najlepiej wykrywaną kategorią są blogi. Licząc średnie odchylenie wskaźników od doskonałości można wywnioskować, że algorytm jest miarodajny w $\approx 92\%$ przypadków. Dowodzi to, że możliwe jest systematyczne kategoryzowanie stron WWW na podstawie ich struktury.